

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau



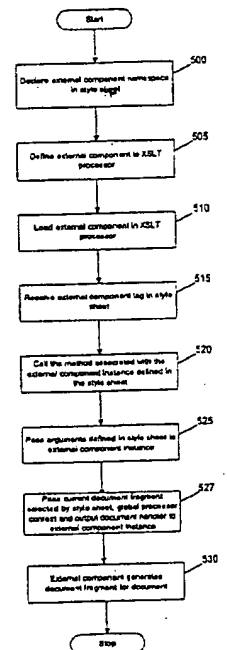
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 17/21, 17/22, 17/30	A1	(11) International Publication Number: WO 00/54174
		(43) International Publication Date: 14 September 2000 (14.09.00)
(21) International Application Number: PCT/US00/06379		(81) Designated States: JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published With international search report.
(22) International Filing Date: 13 March 2000 (13.03.00)		
(30) Priority Data: 60/123,916 12 March 1999 (12.03.99) US Not furnished 10 March 2000 (10.03.00) US		
(71) Applicant: SUN MICROSYSTEMS, INC. [US/US]; 901 San Antonio Road, MS PAL01-521, Palo Alto, CA 94303 (US).		
(72) Inventor: YALCINALP, L., Umit; 1 Debby Lane, Belmont, CA 94002 (US).		
(74) Agents: GARRETT, Arthur, S.; Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., 1300 I Street, N.W., Washington, DC 20005-3315 (US) et al.		

(54) Title: EXTENDING THE CAPABILITIES OF AN XSL STYLE SHEET TO INCLUDE COMPONENTS FOR CONTENT TRANSFORMATION

(57) Abstract

Systems and methods consistent with the present invention use a Namespace paradigm to define an external component reference to a style sheet (500). When the style sheet processor processes the tags in the style sheet, it recognizes the external component declaration (510). The style sheet will contain a name of the external component instance and a definition of the method to execute associated with the external component instance which is executing (520). The XSLT processor then relinquishes control to the external component to execute the method defined in the style sheet (525). The results of the method's execution may be placed in the transform document generated by processing the style sheet (530).



... FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

EXTENDING THE CAPABILITIES OF AN XSL STYLE SHEET TO INCLUDE COMPONENTS FOR CONTENT TRANSFORMATION

RELATED APPLICATIONS

Provisional U.S. Patent Application No. 60/123,916 entitled "Enhanced XML
Processing for Content Transformation" filed March 12, 1999, is relied upon and is
incorporated by reference in its entirety in this application.

BACKGROUND

Field of the Invention

The present invention relates to the use of style sheets in creating a document,
and more particularly, to the use of external components in an XSL style sheet.

Description of the Related Art

Systems' and applications' use of documents has become so prolific that style
sheets are now often used to help manage the documents' display. Style sheets
provide greater flexibility and control over the display of a document's content. XSL
style sheets also allow the content of documents to be transformed, making them as
document transformers where the resulting documents may or may not be used for
display. A user may request a document and the application associated with the
document will use the information contained in the style sheet to display a new
document incorporating the information contained within the style sheet and the
requested document.

Typically, an XSL style sheet includes the use of tags. Tags are codes that
identify an element in the document, such as heading or font, for the purpose of
formatting the document. For example, the tag "<BOLD>" indicates that the text
associated with this tag should appear in bold. Style sheets also, however, have
become more advanced, to include a type of programming language. An example of
this is an Extensible Style Language (XSL) style sheet. XSL is a declarative style
sheet language specified in Extensible Markup Language (XML) which can also be
used to transform XSL documents. The XSL is actually more analogous to a
programming language than to a mechanism designed purely to analyze tags and
display attributes.

With XML, developers may provide functionality by creating their own customized tags. For example, XML supports links that point to multiple documents, as opposed to HTML links, which can reference just one destination each.

XML documents may be served to different clients with varied interests and capabilities. For example, a PC running NETSCAPE may require a document formatted differently than a PDA would. XSL is the style language used by XML to allow different clients to receive different XML documents. XSL is a specification language for separating style from content when creating XML or HTML pages. XSL style sheets allow a single style to be applied to multiple documents.

However, there are limitations to the use of style sheets. An application utilizing a style sheet to display a document often requires that the information contained in the style sheet be application dependent. In other words, the application must be aware of all the tag definitions. In addition, style sheets typically are not very robust and while they use a programming language, do not provide for external calls to components or libraries which may be used to aid in the modification of document information to be displayed.

Similarly, the use of XSL style sheets has been limited. Information contained within the XSL style sheet must be defined within the application and the XSL style sheet itself, and therefore the style sheet becomes application dependent. This sometimes is not desirable when using a style sheet across multiple applications, in which case a separate style sheet for each application would need to be developed. Furthermore, the XSL style sheet is completely self-contained in that no modules or functions can be called outside the style sheet itself. Therefore, what is needed in the art is the ability to create a style sheet that is application independent and to provide a means for creating application dependent functions within that style sheet.

SUMMARY OF THE INVENTION

Methods, systems and articles of manufacture consistent with the present invention overcome the shortcomings of existing style sheets by providing the ability to define components in a style sheet in order to execute methods outside the application. Using a Namespace paradigm, these components, declared within the style sheet, will be recognized by a style sheet processor.

In accordance with one aspect of the present invention, as embodied and broadly described herein, in a data processing system, a method of creating a transform document using a style sheet comprises the steps of receiving a request for an input document, retrieving the style sheet, having tags, associated with the input document, wherein one of the tags represents an external component, processing the tags, including one tag representing an external component, in the style sheet to generate a transform document, and processing a method associated with the external component. An additional step of placing the results of processing the method associated with the external component in the transform document may be performed. The method associated with the external component may generate application specific results. In addition, the style sheet may contain commands written in an XSL format. The step of processing the method associated with the external component may include loading the external component in an XSLT processor and initiating the execution of the method associated with the external component. In addition, the step of validating the style sheet and the input document may be performed. And finally, the step of locating an existing instance of the external component, wherein the processing of the one tag representing an external component includes using the existing instance, may be performed.

In accordance with another aspect of the present invention, as embodied and broadly described herein, in a data processing system, a method for executing an external component in a style sheet, comprising the steps of defining an external component to a style sheet processor, providing a definition of the external component in the style sheet, and processing the external component by the style sheet processor. The steps of loading the external component into the style sheet processor, calling the method associated with the external component defined in the style sheet, and passing an argument contained in the style sheet to the external component for use by the method may also be performed. In addition, the step of loading the external component including the step of locating an existing instance of the external component may also be performed. The external component may be defined to an XSLT processor and the style sheet using a Namespace. And, an application associated with an input document using the style sheet may not recognize the

external component. Processing the external component may include executing a method associated with the external component which generates a result, and placing the result in a transform document.

5 In accordance with yet another aspect of the present invention, as embodied and broadly described herein, a system for processing external components in a style sheet comprises a style sheet containing a definition of an external component and a definition of a method associated with the external component, and a style sheet processor operative to receive a request for an input document, retrieve the style sheet, and call the method associated with the external component defined in the style sheet.
10 The style sheet may contain an argument to pass to the method associated with the external component. In addition, a Namespace may be used in defining the external component to the style sheet and the style sheet processor. The style sheet processor may be an XSLT processor and the style sheet contains tags written in an XSL format. And finally, the style sheet processor may be further operative to generate a transform document and to insert a result of the call to the method associated with the external
15 component.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate presently preferred embodiments of the invention and, together with the general description given above and the detailed description of the preferred embodiments given below, serve to explain the principles of the invention.
20 In the drawings,

Fig. 1 depicts a data processing system suitable for practicing methods and systems in a manner consistent with exemplary embodiments of the present invention;

25 Fig. 2 depicts a more detailed view of an XSLT processor and components related to the XSLT processor suitable for practicing methods and systems in a manner consistent with an exemplary embodiment of the present invention;

Fig. 3 is a flowchart depicting a high level process of generating a transform document using the results of an external call contained within a style sheet;

Fig. 4 is a flowchart depicting the process for generating a transform document containing the result of executing a method associated with an external component; and

Fig. 5 is a method for creating and processing external components in a style sheet.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The following detailed description of the invention refers to the accompanying drawings. Although the description includes exemplary implementations, other implementations are possible, and changes may be made to the implementations described without departing from the spirit and scope of the invention. The following detailed description does not limit the invention. Instead, the scope of the invention is defined by the appended claims. Wherever possible, the same reference numbers will be used throughout the drawings and the following description to refer to the same or like parts.

Introduction

Systems and methods consistent with the present invention use a Namespace paradigm to define an external component to a style sheet. Namespaces are covered in more detail by the "Namespaces in XML" recommendation specification by the World Wide Web Consortium. When the style sheet processor processes the tags in the style sheet, it recognizes the external component declaration and a locator to load the component. The style sheet will also contain a definition of the method to execute associated with the external component and a name which designates a particular instance of a component, and may contain arguments for the method which is executing. The style sheet processor, or the XSLT processor in one embodiment, then relinquishes control to the named external component instance to execute the method defined in the style sheet. The results of the method's execution may be placed in the new document generated by processing the style sheet. However, this is not required, and this execution may provide any results including those not related to the document.

During processing, XSL external components may treat the document specific tags as its input declared by specific Namespaces that designate an input document. The tags that pertain to an external component which are also declared with respect to a particular Namespace will be recognized by the XSLT processor during the validation of a style sheet as declarations of component invocations. During the processing of a document with a style sheet, an XSLT processor will use the component specification along with arguments defined in the XSL style sheet to perform the external component method defined to the XSLT processor associated with the particular component.

The specific instances of a component may be designated by a name. By doing so, different instances of the same component may be active within the same document and useful in performing different tasks. Processing the external component may include executing a method associated with the external component's instance which generates a result and placing the result in a document. The XSLT processor also ensures that a global processing context which is initialized during the validation step by the XSLT processor and can be shared by all the components and their instances referenced within a style sheet during processing of a document is made available to the component instances during processing. The components can utilize the context in their execution and hence change the content of the output document. The components also have access to the current document fragment that is processed by the style sheet and means of generating fragments of a transformed document. The generated result may depend on the context which the XSLT processor supplies to the component's instance. Multiple components or component instances can share the context during an execution.

System Architecture

Figure 1 depicts a data processing system suitable for practicing methods and systems in a manner consistent with exemplary embodiments of the present invention. As shown in Figure 1, application server 104 includes a memory 102, a secondary storage device 112, a CPU 116, an input device 118, and a video display 120. Memory 102 includes application 104, transformation engine 106, registry 108, and XSLT processor 110. Application 104 receives client requests and provides XML

documents to those clients. Transformation engine 106 maintains mapping of all XSL translations, responds to queries from application 104, publishes new registries, and runs an informative servlet. A servlet is a Java applet that runs on a server.

Transformation engine 106 may contain a web interface, Application Program Interface (API), or other input interface. An API is a set of routines, protocols, or tools for communicating with software applications. APIs provide efficient access to the registry without the need for additional software to interface with the file.

Memory 102 contains an XSLT processor 110 that applies the XSL transformation to XML document 106. Secondary storage device 112 contains a registry as XML Document 114. This registry 114 is an XML representation of registry 108. Registry 114 serves as a backup that may be easily loaded into memory 102.

Figure 2 depicts a more detailed view of an XSLT processor and components related to the XSLT processor suitable for practicing methods and systems in a manner consistent with an exemplary embodiment of the present invention. The XSLT processor 205 interfaces with multiple modules in order to process an external component declared within a style sheet. The user 200 may request a document and may provide to the XSLT processor 205 a client type. For example, the user client type might be a PDA or a browser on a PC. The XSLT processor will process this request, and when complete, will send to the user a transform document.

The XSLT processor may perform many functions, several of which are depicted in Figure 2. A validation module 210 will validate the input document request issued by the user and validate the style sheet that is associated with the input document request to ensure that the correct style sheet to generate a transform document is used and valid. The XSLT processor will also understand the external component declarations specified with Namespaces in the style sheets and load them as necessary. The style sheets and component processing module 215 are used to generate a transformed document for the user in response to the document request. This transformed document will incorporate the results of commands given within the style sheet, e.g., tags, and will also incorporate the results of any methods executed by an external component instance, if any. An XML Parser 235 may be used to parse the

input documents and style sheets. An XML Document Builder 240 may be used to build a memory representation of the parsed documents or transformed documents.

To generate this display document, in one embodiment, the XSLT processor 205 accesses style sheets 220, components 225, and content, e.g., XML documents 230. These style sheets 220, components 225, documents 230, may be contained in memory within the data processing system described with respect to Figure 1, in secondary storage or in other storage areas which the XSLT processor 205 can access. As stated, the user will generate a document request. The XSLT processor 205 will recognize, by examining the document request, an associated style sheet. The XSLT processor 205 then retrieves the appropriate style sheet associated with the document requested in order to process and generate a transform document to present to the user.

The components module 225 contains the external components that may be declared in the style sheet by way of a Namespace paradigm. When the style sheet names a reference to an external component, the XSLT processor may create an instance of the component if the named component instance has not been created, or use an already created component instance by looking it up with its name. The XSLT processor may pass any arguments also defined in the style sheet and initiate the specific method execution in the external component instance.

Documents 230 contain input documents, i.e., those documents that contain the content of the document requested by the user, which the XSLT processor will use in conjunction with the style sheet to generate a transform document.

Figure 3 is a flowchart depicting a high level process of generating a new document using the results of an external call contained within a style sheet. This process begins by the user requesting a document having an associated style sheet (step 300). In the present invention, the style sheet does not have to be application dependent. The style sheet may contain XSLT language commands as well as reference to an external component. This external component may be used to generate application specific information or other information desired by the developer of the style sheet. In addition, along with the user request, the client type of the user may be sent along with the document request. The client type will specify what type of client the user is, and will be an indication to the processor processing

the user request how to format the transformed document specific to the client specification.

The method then proceeds to generate a transform document using the style sheet and an external call embedded in an external component in the style sheet (step 305). The external call is to a named component instance and may perform a function defined by the developer creating the external call. In other words, it is not required that the external call perform some function related to the transformed document or even a function related to the document processing. It may be preferable for the developer of the style sheet to insert into the external function parameters and information that may be application dependent. However, as stated above, the external call may also perform various other types of processing.

After processing the external call embedded in the external component, the XSLT processor continues to process the rest of the style sheet and the input document. Other methods of external components embedded within the style sheet might be activated as described above. When the style sheet is completely processed by the XSLT processor, the process sends a transform document containing the results of the external call to the user (step 310). The new document will be transformed or formatted using commands contained in a style sheet associated with the document requested by the user. For example, these commands may define display parameters and other aspects associated with the new document display on the client type for the user. In addition, the style sheet may contain the definition of the external call, as well as any arguments that may be passed through the external call.

Figure 4 is a flowchart depicting the process for generating a transform document containing the result of processing an external component in a style sheet. The process begins by receiving a document request from a user (step 400). The user may request this by using a PC or other type of data processing system by which he has access to the XSLT processor. The XSLT processor receives the document request and client type from the user and validates the document and the style sheet associated with the document (step 405). As part of the validation process, in one embodiment, the XSLT processor validates a document by using an XML parser and

validates a style sheet by using an XML parser and using the XSL language. XSL style sheets are written using the XSL language which is described in XML.

5 The XSLT processor then receives the style sheet containing reference to an external component (step 410). The style sheet may be developed by a user when specifying how a document is to be transformed. The user in the present invention may also specify an external component to the style sheet. The user can also specify a named instance of a component which allows the user to activate different instances of the same external component to process different sections of a document. The naming mechanism that is used as a parameter in the style sheet in referring a component
10 designates a specific instance of an external component. Note that this is different than the Namespaces. Namespaces describe extensions to recognize the component declaration elements in a style sheet. Name attributes within the scope of a component identification element in the XSLT style sheet designate a specific instance of a particular component.

15 The external component may contain a definition of a method which produces results desired by the user. This external component does not need to be application dependent, however, application dependent information may be placed within the external component. The component can also utilize a processing context passed by the XSLT processor.

20 The process then checks to see if an instance of the external component already exists, i.e., is active, in the XSLT processor (Step 412). If it is determined that an instance is not found in Step 413, then the "No" branch is followed and an external component instance is created (Step 414). If it is determined that the instance is found in Step 413, then the "Yes" branch is followed to Step 415. Consequently, the name
25 of the external component may play an important role in this process. If the XSLT processor has encountered this named component instance before, it locates and reuses the same component instance. Otherwise, the processor creates and initializes a fresh new instance and stores its name and the component reference in memory.

30 The style sheet processor begins processing the tags contained within the style sheet to generate the transformed document for the user which may or may not be used for display. If the style sheet contains an external component, as in one

embodiment of the invention, a method will be defined to the external component. The process proceeds by executing the method in the external component instance (Step 415). As part of executing this method, various parameters, i.e., arguments, may be associated with the component that the user has specified in the style sheet.

5 These parameters and the document fragment which is currently being processed by the style sheet will be passed to the external component that is being activated. The XSLT processor, when processing the style sheet, relinquishes control to the instance of the component so that it performs the method defined in the style sheet. After that method has been completed, processing returns to the XSLT processor to continue
10 processing the style sheet and hence the input document.

Once all the tags in the style sheet have been executed, as well as any other method(s) associated with external components contained in the style sheet, the XSLT processor then places in a transform document the results of the tag processing as well as the results of the executed method(s) (step 420). This transform document is now
15 transformed or formatted as desired by the user. It may contain the display characteristics or the transformed document content and tags as intended by the user, as well as the results of any external component processing as defined in the style sheet. Following the placement of the results of the tag processing, including external components, this new document is transmitted to the user using the associated client
20 type (step 425).

Figure 5 is a method for creating and processing external components in a style sheet. This process begins by starting to validate a style sheet and declaring the component Namespace in the style sheet (step 500). This may be performed by defining it as a specific Namespace in the style sheet. The method of declaring
25 Namespaces is well known to those skilled in the art. However, in this instance, the Namespace being declared is a Namespace defining external component activation in a style sheet.

Next, during the validation of the style sheet and processing of the XSLT tags, an external component identified by the specific Namespace declaration is identified
30 and is defined to the XSLT processor (step 505). The use of Namespaces is well known to those skilled in the art. The Namespace paradigm may be used by an

application such that when a specific module resides in the Namespace, the application will be aware of it and will know the location for their retrieval. In the present invention, a component Namespace is defined to an XSLT library so that during the execution of the style sheet, the XSLT processor is aware that an external component is being defined, and it can preload and resolve components which are declared.

Following definition of an external component Namespace to the XSLT processor and after validation of the style sheet is finished, the XSLT processor starts processing an input document with the validated style sheet and the loaded components. It processes the XSL language tags based on the language semantics and transforms the document in the process. When a component declaration needs to be processed in the style sheet during the style sheet processing with the control flow of the XSL language, the XSLT processor looks up the named instance of a component which is specified by the component tag. The component tag also specifies a loading instruction which is handled during the verification step. If the named component instance does not exist, the processor creates a new instance and associates the name specified in the component element with this instance and uses it (Step 510), where it is processed. If the instance already exists, however, the XSLT processor reactivates the named instance. Within the external component tag, in one embodiment, is a method definition associated with the external component instance to be processed.

During the processing of the style sheet, an external component tag is received from the style sheet (Step 515). This is an indication to the style sheet processor, the XSLT processor, that the following text associated with the component tag is for an external component associated with the Namespace declared within that style sheet. This external component tag, along with its Namespace declaration, provides an indication to the style sheet processor that there is an external component whose activation characteristics and its name will be specified, and not to anticipate having a definition for that tag within the application. When the special component tag is encountered, its contents will signal to the XSLT processor to relinquish control so that a method defined to the external component's named instance may be performed.

Next, the method associated with the component is called (step 522). Any values defined in the style sheet for the method received by the XSLT processor are then passed to the specific named method (step 525) as the arguments of the method in the component. The specific method will use the values passed from the XSLT processor as its arguments. This step can use the methods specified in the named external component.

A global context which the XSLT processor passes to the components to utilize and the current document fragment being processed by the style sheet and a transformed document fragment generator are also supplied to the component (Step 527). The methods in the external component instance take all the argument values specified in the method invocation. Three pieces of information need to be available for the external component instance: 1) the document fragment that is currently being processed as selected by the style sheet; 2) the global context that is provided by the XSLT processor; and 3) a means for creating document fragments in the transformed document that is provided to the component instance. The component instance may use or not use the value of the input document fragment or the context during its processing, but should depend on the availability of these items provided by the processor during runtime. This can be handled by different implementation techniques, either by requiring those three additional items to be included in the method signatures by allowing the XSLT processor to pass them to the component instance, or by requiring the component to use a specific API as well as its own API which is specialized with three different methods that pertain to obtaining these three items.

The invention also suggests a specific API which allows the component to obtain the document fragment, means of generating the resulting document fragment, and a global processing context which can be shared among components and their instances. Components which can be activated with this paradigm may utilize an API to be aware of the document processing and the context. For example, a `getInputNode` API may be used to get an input document fragment, a `getDocumentFactory` API may be use to create output document fragment and a `getContext` API may be used global context processing. At runtime, the components

find the required information using the APIs, if necessary. By using the global context, the components can pass information to other components which then may be used in document transformations generated by the components. Components also retain their own state when they are reactivated to allow them to provide intelligence in document transformation.

The XSLT processor is not aware of the functions being performed by processing the external component nor is it in control of these functions. As stated above, processing the external component may include processing associated with the document transformation or display, it may perform other processing associated with the document, or it may even perform processing not associated with the document that the user has defined in the style sheet.

The external component instance, in one embodiment, may then generate a document fragment and place this document fragment into the transform document (step 530). This, of course, presumes that the external component performs a method that, when executed, generates results for the transform document. If no results are generated, the transformed document is not affected by the method execution in the external component instance. When the component instance's method returns execution to the XSLT processor, it resumes processing the document as specified in the style sheet. The XSLT processor during processing the style sheet may require to reuse the named component instance or activate other components instances as specified by the style sheet.

The XSLT processor may also perform error handling. The XSLT processor may record the method invocations if the method signatures in the named component does not match or other internal processing errors might occur. If errors are generated, the component will abort its execution and return a list of exceptions to the XSLT processor. Depending on the severity of the errors, the XSLT processor may log this as an error and continue processing to generate a transformed document, or the errors severity may abort the XSLT processor. Severe errors include, but are not limited to, unloadable components, inability to activate component instances and methods, or components indicating severe errors.

When the XSLT processor completes the processing of the style sheet, the new document containing the results of the style sheet along with the results of all external component instances is placed in the new document and sent to the user. It will be recognized by those skilled in the art that while the foregoing description included a discussion of one external component being executed in the style sheet, multiple external components may be executed and defined in a style sheet as defined by the user.

An Example of an External Component in a Style Sheet

Below is an example of a style sheet written in XSL:

```
10 <?xml version="1.0"?>
    <!--XSL Style sheet, DTD omitted -->
    <xsl:stylesheet
        xmlns:xsl="http://www.w3.org/TR/WD-xsl"
        xmlns:xslcomponent="http://www.javasoft.com/lhs"
15     >
        <xsl:template match="paragraph">
            <xslcomponent:component
                name="summary1"
                class="com.sun.lhs.impl.samplecomponents.SummaryBuilder"
20             >
                <xslcomponent:args>
                    <xslcomponent:methodname="genSummary"/>
                    <xslcomponent:argname="bufferize" value="100"/>
                >
                </xslcomponent:args>
25             </xslcomponent:component>
            <xsl:apply-templates/>
        </xsl:template>
    </xsl:stylesheet>
```

30 The above example includes a Namespace declaration,
xmlns:xslcomponent=http://www.javasoft.com/lhs,

a component tag signifying that an external component will be used,

`xslcomponent:component`

a definition of a instance name which designates the specific instance of the component that will execute,

`name = "summary1"`

a class tag which describes the specific component to be loaded, in one embodiment, i.e., the class tag describes the full class name for components written in the Java language and arguments to pass to the external component to execute the method. In the example below, the method to be executed is named "genSummary" and it has one argument, the value 100),

```
<xslcomponent:args>
```

```
<xslcomponent:methodname="genSummary"/>
```

```
<xslcomponent:argname="bufferize" value="100"/>
```

```
>
```

```
</xslcomponent:args>
```

Conclusion

Systems and methods consistent with the present invention use a Namespace paradigm to define an external component reference to a style sheet. When the style sheet processor processes the tags in the style sheet, it recognizes the external component declaration. The style sheet will contain a name of the external component instance and a definition of the method to execute associated with the external component instance, and may contain arguments for the method which is executing. The XSLT processor then relinquishes control to the external component to execute the method defined in the style sheet. The results of the method's execution may be placed in the transform document generated by processing the style sheet. However, this is not required and other results may be generated, including those not related to the transform document.

The foregoing description of an implementation of the invention has been presented for purposes of illustration and description. It is not exhaustive and does not limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing of the

invention. For example, the description refers to an external component instance, which will be recognized by those skilled in the art as an instantiation of an object in a system using object oriented language processing. However, other systems using other language processing mechanisms may be used. As yet another example, the described implementation includes software, but one embodiment of the present invention may be implemented as a combination of hardware and software or in hardware alone. The invention may be implemented with both object-oriented and non-object-oriented programming systems. Additionally, although aspects of the present invention are described as being stored in memory, those skilled in the art will appreciate that these aspects can also be stored on other types of computer-readable media, such as secondary storage devices, like hard disks, floppy disks, or CD-ROM; a carrier wave from the Internet or other propagation medium; or other forms of RAM or ROM. The scope of the invention is defined by the claims and their equivalents.

WHAT IS CLAIMED IS:

1. In a data processing system, a method of creating a transform document using a style sheet, comprising the steps of:
receiving a request for an input document;
5 retrieving the style sheet, having tags, associated with the input document, wherein one of the tags represents an external component;
processing the tags, including one tag representing an external component, in the style sheet to generate a transform document; and
processing a method associated with the external component.
- 10 2. The method of Claim 1 further comprising the step of placing the results of processing the method associated with the external component in the transform document.
3. The method of Claim 1, wherein the method associated with the external component generates application specific results.
- 15 4. The method of Claim 1, wherein the style sheet contains commands written in an XSL format.
5. The method of Claim 4, wherein the step of processing the method includes loading the external component in an XSLT processor and initiating the execution of the method associated with the external component.
- 20 6. The method of Claim 1 further comprising the step of validating the style sheet and the input document.
7. The method of Claim 1 further comprising the step of locating an existing instance of the external component, wherein the processing of the one tag representing an external component includes using the existing instance.
- 25 8. A method for executing an external component in a style sheet, comprising the steps of:
defining an external component to a style sheet processor;
providing a definition of the external component in the style sheet; and
processing the external component by the style sheet processor.
- 30 9. The method of Claim 8 further comprising the steps of:
loading the external component into the style sheet processor;

calling the method associated with the external component defined in the style sheet; and

passing an argument contained in the style sheet to the external component for use by the method.

5 10. The method of Claim 9, wherein the step of loading the external component includes the step of locating an existing instance of the external component.

10 11. The method of Claim 9 further comprising the steps of:
naming a first and a second instance of the external component; and
processing the first and the second instance of the external component;

12. The method of Claim 10 further comprising the step of processing different methods within the existing instance of the external component.

13. The method of Claim 8 wherein the external component is defined to an XSLT processor and the style sheet using a Namespace.

15 14. The method of Claim 8, wherein the step of processing the external component includes executing a method associated with the external component which generates a result and placing the result in a transform document.

20 15. The method of Claim 8 further comprising the step of providing a global context, a current input document fragment as processed by the style sheet and a mechanism to generate a transformed document fragment to the external component.

25 16. The method of Claim 8 further comprising the steps of
retaining state information from the processing of the external component;
reusing the state information during the processing of a second activation of the external component, wherein a result generated by the processing of an external component depends on the state information.

17. A system for processing external components in a style sheet, comprising:

a style sheet containing a definition of an external component and a definition of a method associated with the external component; and

a style sheet processor operative to receive a request for an input document, retrieve the style sheet, and call the method associated with the external component defined in the style sheet.

5 18. The system of Claim 17, wherein the style sheet contains an argument to pass to the method associated with the external component.

19. The system of Claim 17, wherein a Namespace is used in defining the external component to the style sheet and the style sheet processor.

20. The system of Claim 17, wherein the style sheet processor is an XSLT processor and the style sheet contains tags written in an XSL format.

10 21. The system of Claim 17, wherein the style sheet processor is further operative to generate a transform document and to insert a result of the call to the method associated with the external component.

22. A computer-readable medium containing computer-executable instructions for creating a display document using a style sheet to perform a method
15 comprising the steps of:

receiving a request for an input document;

retrieving the style sheet, having tags, associated with the input document, wherein one of the tags represents an external component;

20 processing the tags, including one tag representing an external component, in the style sheet to generate a transform document; and
processing a method associated with the external component.

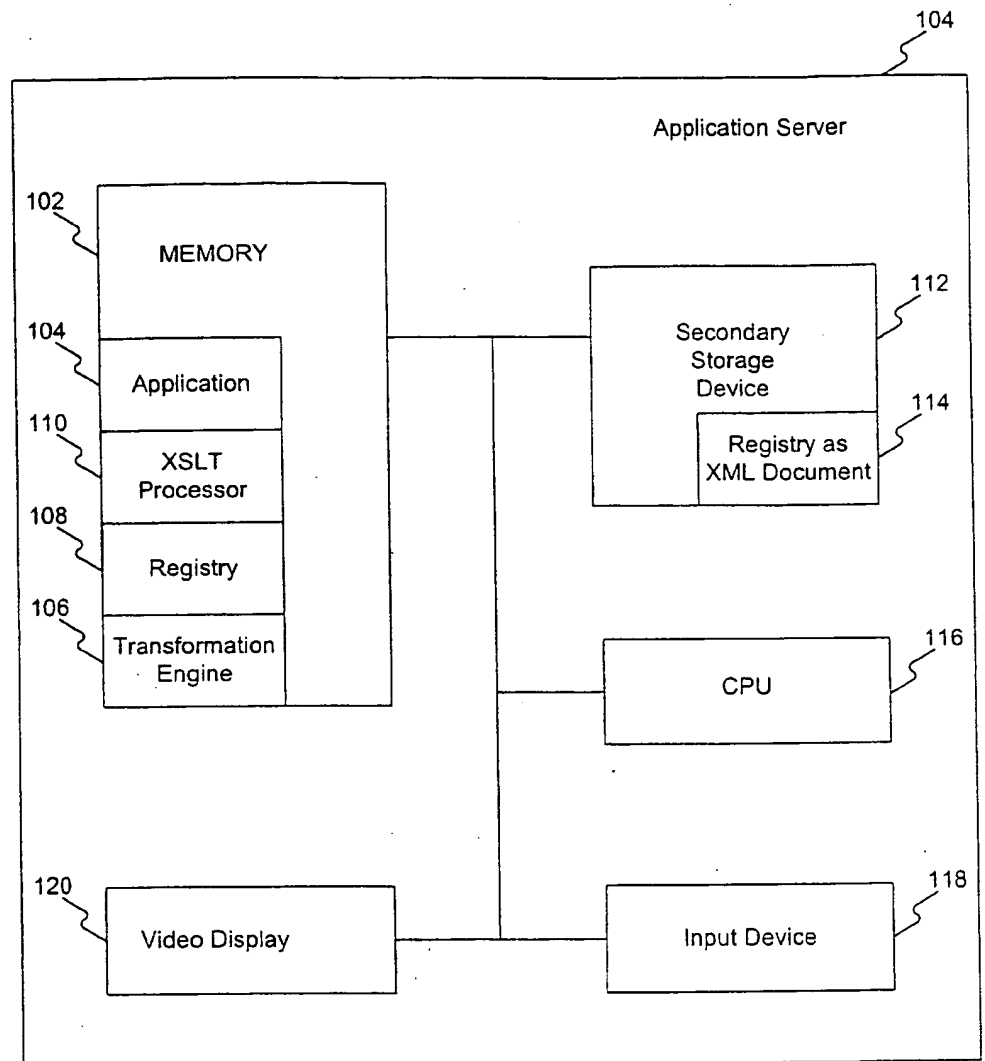


FIG. 1

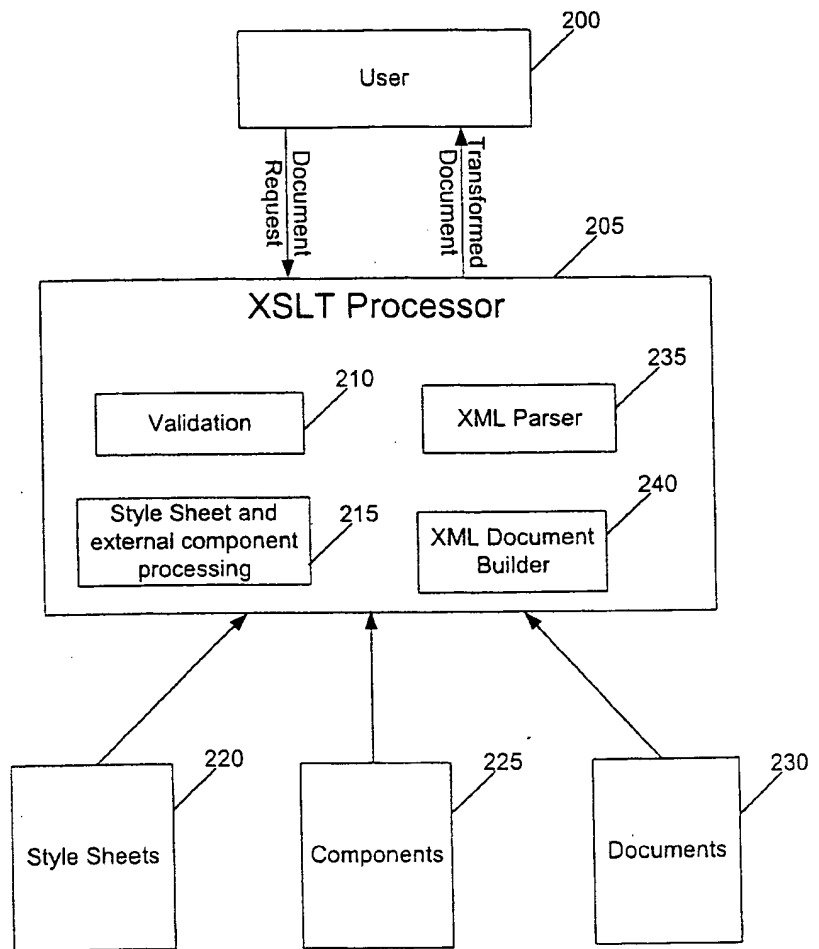


FIG. 2

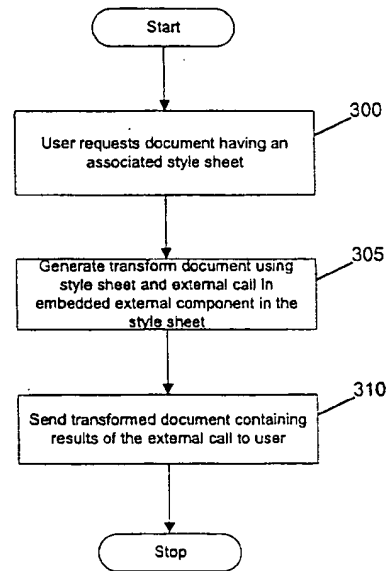


FIG. 3

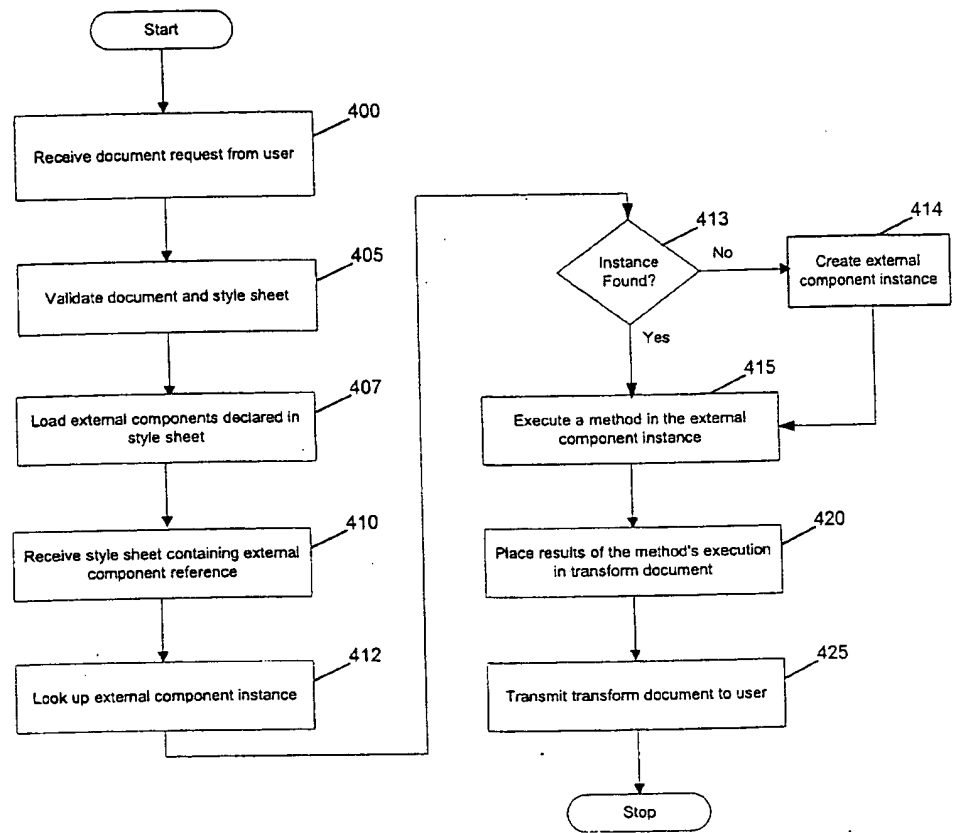


Fig. 4

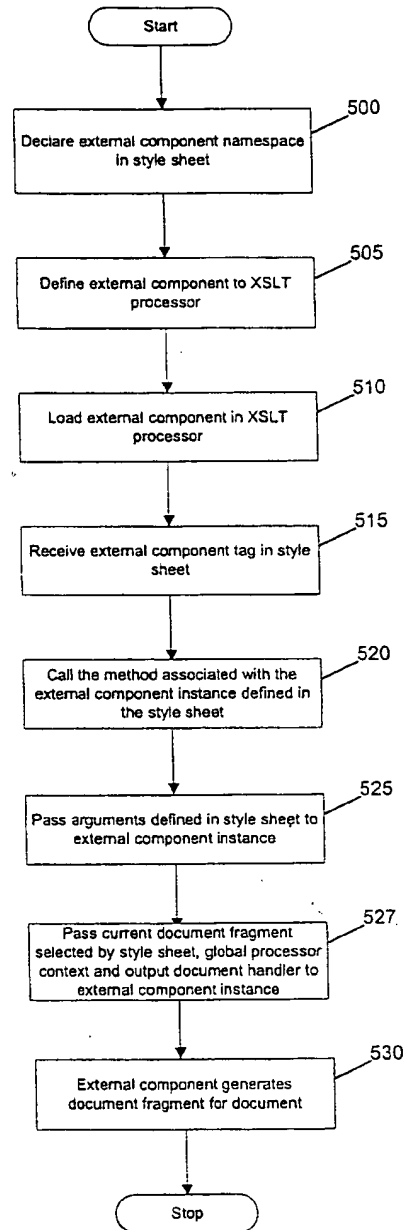


Fig. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/06379

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 17/21, 22, 30

US CL : 707/501, 513, 517

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 707/501, 513, 517

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
WEST US Patent database; DIALOG database

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WALSH, N. The Extensible Style Language: XSL, WEB Techniques, January 1999, vol.4, no.1, p.49-50, 52, 54-55.	1-22
A	EXNER, N. Examining XML: new concepts and possibilities in Web authoring, Computers in Libraries, November 1998, v18, n10, pp.50-53.	1-22
A	STEVENSON, M. Xtending the Enterprise, Intelligent Enterprise, November 1998, pp.8.	1-22
A	STANEK, W. XML 201, PC Magazine, October 1998, v17, n17, pp.217.	1-22
A	SENNA, J. XML bridges the gap, InfoWorld, June 1998, v20, n22, pp. 88.	1-22



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:	* I* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
* A* document defining the general state of the art which is not considered to be of particular relevance	* X* document of particular relevance, the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
* E* earlier document published on or after the international filing date	* Y* document of particular relevance, the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
* L* document which may throw doubts on priority claim(s) in which is cited to establish the publication date of another citation or other special reason (as specified)	
* O* document referring to an oral disclosure, use, exhibition or other means	* &* document member of the same patent family
* P* document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

29 APRIL 2000

Date of mailing of the international search report

25 MAY 2000

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

STEPHEN HONG

Telephone No. (703) 305-3900

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/06379

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	ZEICHICK, A. XML and XSL, Network, November 1998, pp.23.	1-22
A	MCGRATH, S. Rendering XML documents using XSL, Dr. Dobb's Journal, v23, n7, pp.86(6)	1-22
A, P	US 6,031,989 A (CORDELL) 29 February 2000, ALL	1-22
A	US 5,860,073 A (FERREL et al) 12 January 1999, ALL	1-22
A, P	US 6,012,098 A (BAYEH et al) 04 January 2000, ALL	1-22
A, P	US 6,023,714 A (HILL et al) 08 February 2000, ALL	1-22
A	US 5,870,549 A (BOBO, II) 09 February 1999, ALL	1-22

THIS PAGE BLANK (USPTO)